Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

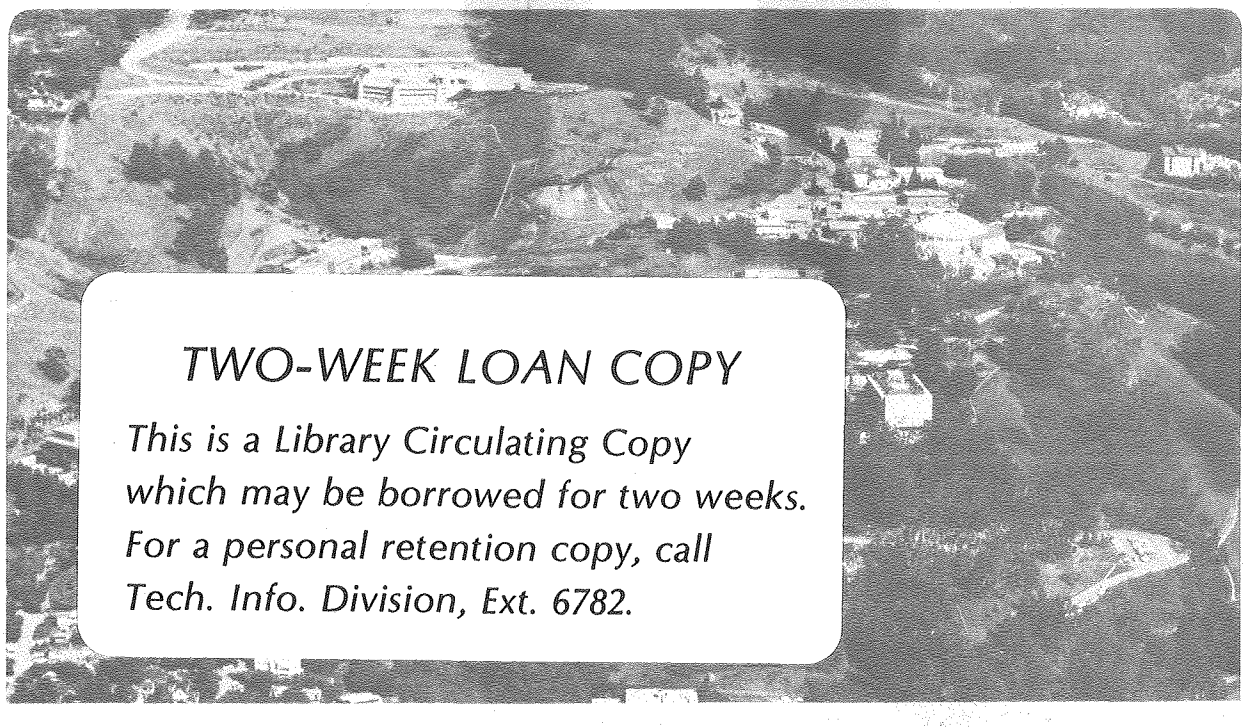
Engineering & Technical
Services Division

To be presented at the Texas Instruments-Members' Information Exchange, 1981 National Symposium, New Orleans, LA, March 8-11, 1981

POWER BASIC AND THE 9980/9981

John Meng

March 1981



Prepared for the U.S. Department of Energy under Contract W-7405-ENG-48

DISCLAIMER

POWER BASIC AND THE 9980/9981

John Meng
Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720 U.S.A.

Abstract

Microprocessors are today properly classified as logic workhorse devices, falling into sequence with gates and flip-flops just a few years ago and with the diode, transistor and discrete component arrays which served before. To be effectively used as a component the micro must be small, easy to use, flexible and inexpensive. We are building self-contained miniature TI 9980/9981-based modules (2 1/2" x 6" pc board) which are capable of executing power basic and which feature one RS-232 port and 1024 bytes of random access memory (RAM) for program storage. A logical expansion of this module into an array of several such modules on a single larger circuit card is practical. The multiple modules share access to a single Power Basic ROM set and may be made to share areas of RAM as well. Special considerations necessary for the successful use of Power Basic with 9980/9981 central processors are described in detail.

Introduction

The desire to use a microprocessor system as a component is not new, and has stimulated the development of several computer on-a-chip devices, including among others the TMS 9940. However, the compromises demanded to make such devices physically buildable exclude them from many fairly sophisticated applications. For example, memory size restrictions tend to limit the use of high level languages, and to restrict program size and the size of data storage areas. On the other hand, many applications do not demand the full power of available single or multiple-board systems. We have successfully applied TMS 9980/9981 CPU's executing Power Basic programs to each of four such applications during the past year.

This work was supported in part by the Environmental Protection Agency under Interagency Agreement with the Assistant Secretary for Environment, Office of Health and Environmental Research, Pollutant Characterization and Safety Research Division of the U.S. Department of Energy under Contract No. W-7405-ENG-48.

Applications

First, we required a simple bench tester to be used to check modules which are being built into a large data analysis system. The tester had to be easily usable and reconfigurable by the technicians doing the testing. It also had to be flexible enough to become the tester for a sequence of five logically different units. Since the data analysis system is a one-of-a-kind device, we could not afford extensive development effort for this test module. Our solution was a 9980-based module running Power Basic. We have added an array of TMS 9901 input/output chips to provide us with the necessary signals to do the testing.

The next challenge was a controller for an incremental plotter to produce hard copies of graphic displays normally transmitted to a storage CRT. The controller is required to intercept and buffer ASCII sequences of data transmitted over an RS-232 link. It then converts the code sequences into single-step movement commands and executes these on the plotter. Once again we applied the 9980 module. The translation from ASC II to plotter code and the actual plotter drive are done in Basic. The RS-232 input drives an interrupt which invokes a short assembly-language program used to fill the character buffer. The CPU 9980 board has been augmented by one TMS 9902 RS-232 chip to receive input and by one TMS 9901 parallel input/output chip to drive the plotter. The Power Basic program initializes the assembly language program and controls the interrupt enable and disable functions. The Basic source program is burned into ROM and preset to automatically start following a hardware reset. The reset is generated during a power-up operation.

Another 9980 CPU resides within the bowels of a large data analysis system we are building. It is necessary for data paths to be switched very quickly in this system, and to be switched in a predefined sequence. Elements of the sequence must be changeable on command at a slower rate. To handle the actual switching and the storage of proper sequences, we are using first-in-first-out (FIFO) stacks. The preloading of the proper switching sequence into these stacks is achieved with a 9980 controller module. Although the switching must

be done very rapidly on demand, the overall rate at which switching occurs is quite low, so the relatively low speed at which the 9980 preloads the FIFO lists is adequate.

Finally, we are using a 9980 microcontroller to oversee the mechanical operations necessary to operate a large disk drive. The 9980 controller acts as a slave to a smarter higher-speed controller (one based on the 990/101 16-bit single card processor). The 9980 steps heads, selects heads, monitors hardware status bits and initiates reads and writes.

The Microcontroller

The microcontroller is configured as shown in Figure 1. Four TMS 2532 EROMS are used to store Power Basic and the user program. Four TMS 4045 RAMs implement volatile storage. We are using the clock generator chip built for use with TMS 9900 systems to generate the 9980 clock. ROM, RAM and input/output chip selects are generated from a single 8 x 1k schottky ROM. Expanding the controller requires adding input/output chips (TMS 9901's and/or TMS 9902's and/or other devices) and an additional ROM for producing chip selects.

When we first looked into the possibility of utilizing the 9980/9981 running Power Basic, only one problem was immediately evident. The 9980 is capable of operating only up to 2.5 MHz, slower than the 3 MHz which is normal for TMS 9900 based systems. Consequently, a baud-rate code sent to the TMS 9902 RS-232 chip would not set the same rate on a 9980 system that it would set on a 9900 system operating at 3 MHz. Not having access to Power Basic source code, it was not initially practical to think of modifying the Power Basic EROMS to accommodate this difference. As a result, at first we operated our 9980/9981 controllers at 1.5 MHz, exactly half the frequency of a 9900 system. At start-up, Power Basic apparently detects the BAUD rate of the terminal device by counting instructions executed during the first bit input. It then sends the appropriate baud-rate code to the TMS 9902. Executing at half speed, the count will be half of what it should be, falsely telling the program that the BAUD rate is twice what it actually is. The program then

sets the 9902 to transmit and receive at double the appropriate rate. Since we are driving the 9902 at 1.5 MHz from the ϕ_3 output on the 9980/9981 chip, the resulting BAUD should in fact be correct. Three things complicate this apparently happy conclusion, however.

First, Power Basic does not support 600 BAUD. Consequently, a 300 BAUD terminal connected to the 9980 does not respond properly. Unfortunately, 300 BAUD is very common in printing terminals, and this faulty response is a severe handicap. Second, a 1200 BAUD terminal will respond correctly, but the Power Basic start-up routine believes it to be 2400 BAUD. The result is that the end-of-line and inter-line pad characters required for some 1200 BAUD printing terminals (such as the silent 700) are missing, although the character transmission rate is correct.

Our original solution for these unfortunate circumstances was to start the system with a 9600 BAUD video terminal and execute:

```
BASE 80H:: CRB[11] = 1:: CRB[12] = 1:: CRF(13) = 468H
```

thereby switching the terminal 9902 to 300 BAUD, leaving the 9980 thinking it was running at 9600 BAUD. The lack of end-of-line spacers still proved an annoyance, however. We found that by changing the last part of the above line to:

```
::CRF(13) = 49CH
```

We could operate satisfactorily at 200 BAUD. Subsequently, via the TI Power Basic Hot Line, we located and changed the BAUD rate table within our copy of Power Basic and are now operating correctly with a start-up at 300 BAUD, and the capability of using the BAUD command supported by Power Basic to correctly change BAUD rates.

Second, the real time clock, when implemented on this 9980/9981 system (by adding a TMS 9901), runs at half speed. It is possible to use the same trick we used on the TMS 9902 RS232 chip to nearly correct the time-of-day clock.

That is, we can access the interval timer with BASIC commands and thereby change its time interval. Unfortunately, this does not quite work. It is necessary to load a value into the 9980/9981 timer which is exactly half that of the 9900 timer, thereby counting down half the distance to produce the same time interval at half the clock frequency. The value preloaded into the timer for the 9900 system is an odd number, producing a rounding error in the 9980/9981 system. Our solution to this problem is to change the INC instruction in the clock interrupt service routine to an INCT instruction, counting twice for each tic rather than once. We were able to find the location of this instruction via the interrupt pointer in low memory.

One More Bug

The module as we have described it so far is in successful use in the applications described earlier. However, upon using it more extensively, an annoying bug became evident. The PRINT statement works with a single argument, and under some circumstances with multiple arguments. However, the statement: PRINT 1;1;1; not only refuses to print but hangs the BASIC interpreter in a short loop from which there is no escape but a general reset. Our controllers are all working, however, and none of them demanding printing during actual use. A thorough search of the available printed data on the 9900 chip and on the 9980/9981 chips turned up no clues. The TI Power Basic Hot Line had no information on 9980/9981 Power Basic execution. It seemed most unlikely that any fundamental difference between the two device types could result in such a limited problem. The successful operation of four controllers seemed to confirm this assertion.

We were, of course, wrong. There is a very fundamental difference between the 9900 and the 9980/9981; a difference never explicitly covered in the printed material. The program counter in the 9900 always contains even numbers. In the 9980/8891, the program counter apparently steps in bytes and can, in fact, contain odd numeric values. A Branch and Link to subroutine (BL) command executed on a TMS 9900 results in an even value being stored in register 11. The same instruction executed on a TMS 9980 can result in an odd number being stored in linkage register 11. Under most circumstances, this is irrelevant. However, if the subroutine uses register 11 as a byte pointer to

an argument list in the calling routine, the answer will be different depending on the chip used for execution. Upon discovering this, we used Power Basic to search itself for byte operations with register 11, and found three potential trouble spots. Appropriate patches cleared up our PRINT problems.

The Multicontroller

A logical expansion of the microcontroller [Fig. 1] is the multicontroller [Fig. 2]. The multicontroller is a microcontroller reproduced several times on a single logic plane. Economies are possible through the time shared use of a single Power Basic ROM set by all the 9980's on the one large circuit card. If the application demanded it, RAM could be common to all the micro-units within a multicontroller either to facilitate communication or to supply a data base for multiple parallel computations.

Such a system could be used to supply independent CPU's to an entire classroom of students. Another possible use is in a situation where massive amounts of parallel computation are needed on common data bases. Monte Carlo calculations may lend themselves to such a system of a large array of independent processors, for example.

Conclusions

The gap between single-chip microcomputers and single-board microcomputers is sufficiently wide to permit the advantageous entry of a mid-range system capable of executing a full Power Basic. The 9980/9981 is slower than the single-board 9900-based microcomputer, but still fast enough to be used in a wide range of mechanical control environments. Inexpensive arrays of these 9980-based microcontrollers can be assembled to be used in education or in other environments requiring large numbers of simultaneous executions at a very modest cost.

Acknowledgment

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

This work was supported in part by the Environmental Protection Agency under Interagency Agreement with the Assistant Secretary for Environment, Office of Health and Environmental Research, Pollutant Characterization and Safety Research Division of the U.S. Department of Energy under Contract No. W-7405-ENG-48.

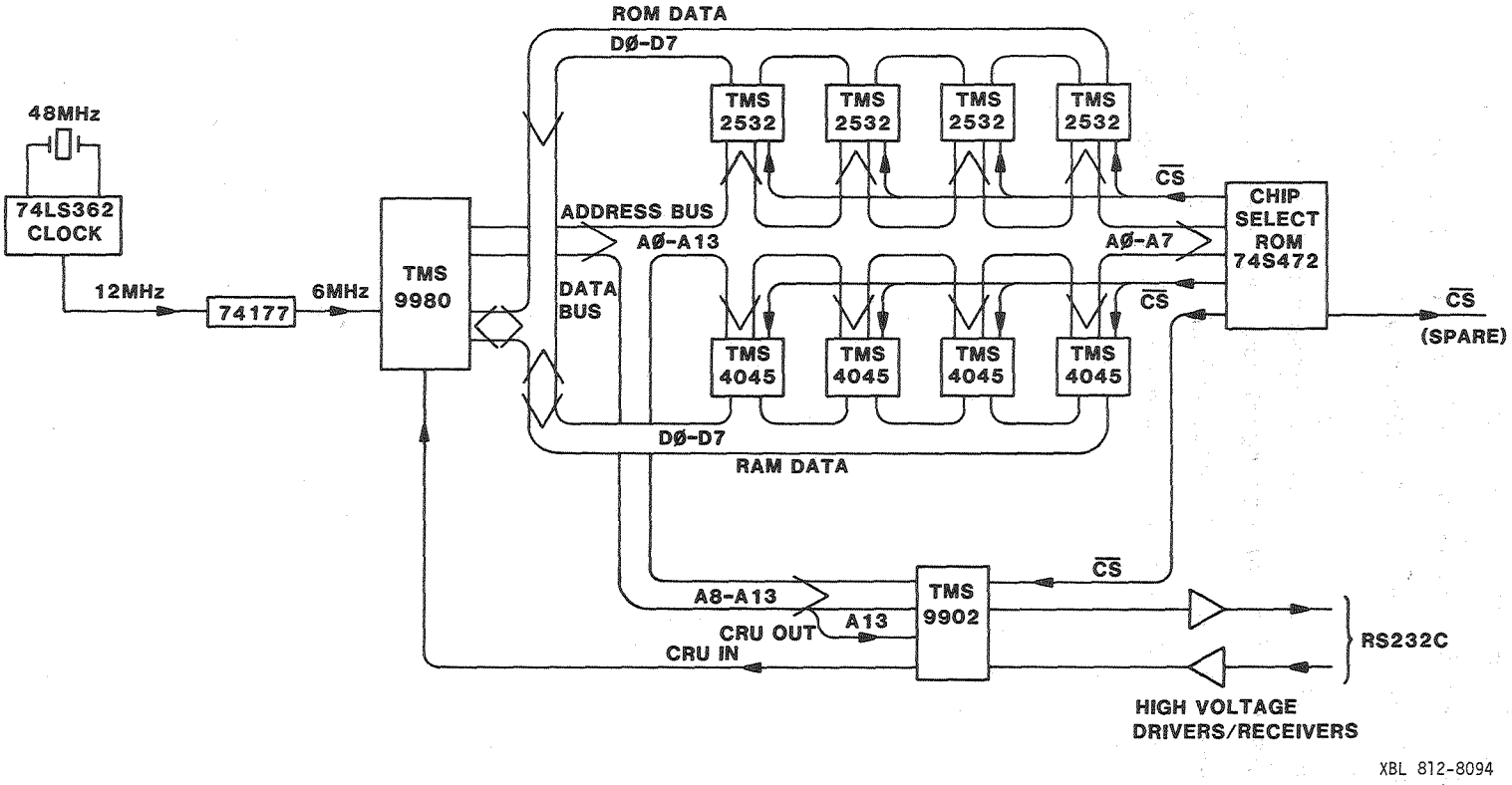


FIGURE 1

The microcontroller module fits onto a 2" x 6" logic substrate and is capable of executing Power Basic. Additional chip select ROM's may be added for input/output expansion and/or RAM expansion.

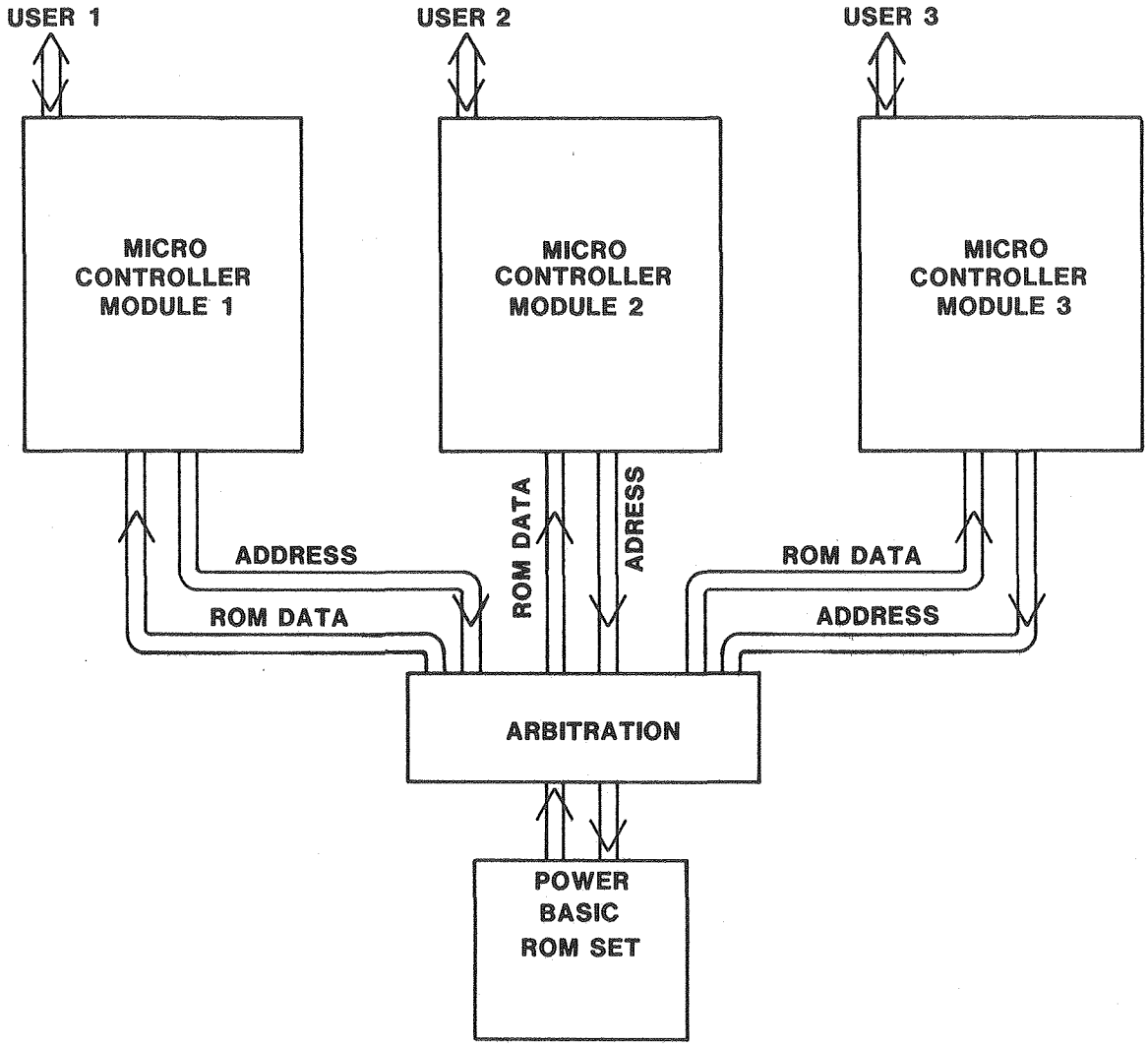


FIGURE 2

The Multicontroller is a sequence of microcontrollers sharing one set of Power Basic ROM's, the most expensive part of each microcontroller. The system features complete user independence and parallel execution at modest cost.